Merging Flows in Terminal Manoeuvring Areas via Mixed Integer Linear Programming

Mohammed Sbihi, Marcel Mongeau, Ji Ma, Daniel Delahaye

Proceedings of GOW'16, pp. 113 – 116.

Merging Flows in Terminal Manoeuvring Areas via Mixed Integer Linear Programming

Mohammed Sbihi[1], Marcel Mongeau[1], Ma Ji[1] and Daniel Delahaye[1]

[1]*ENAC- Université de Toulouse, 7 avenue Edouard Belin, 31055 Toulouse cedex 4, France*
[mohammed.sbihi|marcel.mongeau|ji.ma|daniel.delahaye]@recherche.enac.fr

Abstract This paper focuses on the aircraft merging and sequencing problem at Terminal Manoeuvring Areas through the use of Controlled Time of Arrival (CTA). A Mixed-Integer Linear Programming formulation is proposed in order to minimize the number of non achievable CTAs while maintaining separation between aircraft with regard to the horizontal, wake-turbulence, and runway occupancy time constraints. Computational experiments performed on real-world case studies of Paris Charles De-Gaulle (CDG) airport show that the approach is viable.

Keywords: Aircraft merging and sequencing, Controlled time of arrival, Optimization model, MILP.

1. Introduction

The Terminal Manoeuvring Area (TMA) is a designated area of controlled airspace surrounding one or several airports where there is a high volume of traffic. It is designed to handle aircraft arriving to and departing from airports. TMA is identified by many researchers as one of the most critical parts of the air transportation system. Therefore, there is a need for improving efficiency and increasing capacity by using efficient approaches and algorithms. In this paper, we focus on the problem of merging flight flows into the TMA. During the transition from the en-route to the terminal airspaces, aircraft arriving from different entry points must be merged and organized into an orderly stream while maintaining a safe separation between them. In moving to the future SESAR concept of Trajectory Based Operations, Air Traffic Controllers (ATC) can merge arrival traffic streams into sequences by the use of so-called *Controlled Times of Arrival* (CTA) at the TMA entry point, called *Initial Approach Fix* (IAF). CTA can be achieved using the airborne *Required Time of Arrival* (RTA) functionality, a feature of modern Flight Management Systems designed to calculate and adjust the speed of the aircraft to arrive at a given point in space at a defined target time. CTAs are determined by ATC (typically using an arrival manager tool) and set when the aircraft is around 150-200NM from touchdown. Such calculations might take into account, among other things, downlinked aircraft *Estimated Time of Arrival* (ETA) (or a time-window $[\mathrm{ETA}_{\mathrm{min}}, \mathrm{ETA}_{\mathrm{max}}]$). In [1], De Smedt *et al.* investigated the application of RTA to a real sequence of arriving aircraft into Melbourne, Australia. They found that pressure on the terminal area would sometimes require aircraft to lose more time than what is possible through the RTA capability, and hence require additionally a recourse to other conventional sequencing techniques to provide a sequence resolution.

In this paper, we consider the problem of assigning CTAs to arriving aircraft in order to reduce the number of CTAs that fall outside the $[\mathrm{ETA}_{\mathrm{min}}, \mathrm{ETA}_{\mathrm{max}}]$ windows subject to operational constraints related to wake turbulence, horizontal separation, and runway occupancy time. This problem is very close to the problem of minimizing the number of late jobs on one machine, which is known to be NP-hard in the strong sense [2]. We propose a Mixed-Integer Linear Programming (MILP) formulation of this problem and report computational experiments on real-world case studies from Paris CDG airport using Gurobi optimization solver.

2. MILP formulation

2.1 Given data

In TMA the traffic is arranged so that there is a basic segregated and separated Terminal Area flight path structure with arriving traffic coming through one of a number of IAF. For example, Figure 1 displays the arriving network structure at Paris CDG airport to runway 26L. In this arrival procedure, four routes, originating from IAF MOPAR, LORNI, OKIPA and BANOX, fuse into one single route towards the runway.

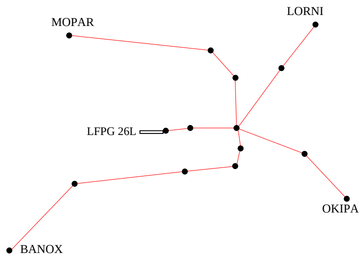

Figure 1. The route structure model for LFPG runway 26L.

We model the route structure as a graph, or to be more precise as a tree here, $G = (V, E)$ in which the aircraft are allowed to fly. The vertex set V is the set of way-points and E is the set of the arcs interconnecting these way-points by a straight-line segments . The arcs are assigned the natural orientation towards the root r, which plays the role of the runway threshold. Each other leaf $e \in V$ corresponds to an entry point and admits a unique corresponding path (route) r_e leading to the root. We are given a set of flights (or aircraft), $\mathcal{F} = \{1, \ldots |\mathcal{F}|\}$, and for each flight $f \in \mathcal{F}$ the following data is also given:

- e_f: entry way-point at TMA (this determines the route $r_f := r_{e_f}$ of flight f)
- t_f: ETA at the entering point $e_f \in V$
- s_f^u: speed (supposed constant) of f on the arc $u \in E$
- $[t_f - \underline{\Delta}_f, t_f + \overline{\Delta}_f]$: the [ETA$_{\min}$,ETA$_{\max}$] window.

2.2 Optimization variables

For each flight f we associate the continuous variable x_f, representing its assigned CTA, and the binary variable y_f indicating whether the assigned CTA falls outside the [ETA$_{\min}$,ETA$_{\max}$] window ($y_f = 1$) or not ($y_f = 0$). Considering two flights f and g, we have to decide which lands first. Thus, we further introduce the decision variable

$$\delta_{f,g} = \begin{cases} 1 & \text{if } f \text{ lands before } g \\ 0 & \text{otherwise} \end{cases}$$

Remark that $\delta_{f,g}$ decides also the passing order on any node $v \in r_f \cap r_g$. We also introduce auxiliary variable t_f^v, representing the passing time through node v. It is connected to x_f by $t_f^v = x_f + \sum_{u \in r_f^v} \frac{d_u}{s_f^u}$, where the r_f^v contains the arcs of r_f before v, and d_u is the length of u.

2.3 Objective function

The aim is to minimize the total number of non achievable CTAs:

$$\sum_{f \in \mathcal{F}} y_f. \tag{1}$$

2.4 Constraints

The first set of constraints indicates the decision interval for the CTA. Ideally we would ask each x_f to be an achievable CTA, *i.e.* $t_f - \underline{\Delta}_f \leq x_f \leq t_f + \overline{\Delta}_f$. However, such a requirement, will in general render the problem infeasible. Thus, at the price of possibly invoking other conventional sequencing techniques, we relax this constraint by

$$t_f - \underline{\Delta}_f \leq x_f \leq t_f + R_{\max}, \qquad f \in \mathcal{F} \tag{2}$$

where $R_{\max}$ is the maximum moving-backward value of CTA, a user-defined parameter.

Now, considering pairs (f, g) of flights, we have that

$$\delta_{fg} + \delta_{gf} = 1, \qquad f, g \in \mathcal{F}, g > f. \tag{3}$$

In words, either flight f must land before ($\delta_{fg} = 1$) or after ($\delta_{gf} = 1$) flight g. It is trivial to see that, for certain pairs (f, g) of flights, we can decide whether $\delta_{fg} = 1$ or whether $\delta_{gf} = 1$ based on the particular input data in a preprocessing step. The link between x_f and y_f is given by

$$x_f - y_f(R_{\max} - \overline{\Delta}_f) \leq t_f + \overline{\Delta}_f, \qquad f \in \mathcal{F}. \tag{4}$$

Indeed, if the CTA is not achievable, then this constraint implies that $y_f = 1$. Otherwise, both values $y_f = 0$ and $y_f = 1$ are feasible, but in the minimal solution y_f will necessarily be 0.

Operational constraints. In this problem, we consider three separation requirements.

Runway separation constraints. For each ordered pair of flights (f, g) a minimum separation of $\tau_{f,g}$ units must be maintained between the landing times t_f^r and t_g^r of f and g. This minimum separation $\tau_{f,g}$ depends on the wake turbulence categories of f and g. This separation is insured by the following constraint (M denotes a sufficiently large positive constant)

$$t_g^r - t_f^r \geq \tau_{f,g} - (1 - \delta_{f,g})M, \qquad f, g \in \mathcal{F}, f \neq g. \tag{5}$$

Weak-turbulence constraints. For each pair of successive aircraft (f, g), the International Civil Aviation Organization regulates the minimum spacing between them to avoid the danger of wake turbulence. It is a distance-based separation $w_{f,g}$. As the speed is assumed to stay constant throughout one arc, it is sufficient to check this separation constraint at the nodes $v \in r_f \cap r_g$. This is achieved by imposing the following constraint

$$t_g^v - t_f^v \geq \max\left(\frac{w_{f,g}}{s_g^{u_{g,v}^-}}, \frac{w_{f,g}}{s_f^{u_{f,v}^+}}\right) - (1 - \delta_{f,g})M, \qquad v \in r_f \cap r_g; f, g \in \mathcal{F}, f \neq g \tag{6}$$

where $u_{g,v}^-$ (resp. $u_{f,v}^+$) is the arc of r_g incoming to (resp. the arc of r_f outgoing from) node v.

Horizontal separation constraints. Aircraft must satisfy a minimum given horizontal separation, d_h, based on radar (typically $d_h = 3$ NM in the TMA). In order to give a necessary and sufficient condition for the horizontal separation, we need the following assumptions:

(H1) The distance between any two non-adjacent arcs u_1 and u_2 is greater than or equal to d_h.

(H2) For any two distinct adjacent arcs $u_1 = (v_1, v)$ and $u_2 = (v_2, v)$ (or $u_2 = (v, v_2)$), the distance between v_2 and the line segment $[v_1, v]$ and the distance between v_2 and the line segment $[v2, v]$ are greater than or equal to d_h.

Assumption (H1) implies that a *conflict* (the distance between two aircraft is less than d_h) can only occur between two aircraft flying on the same arc or two adjacent arcs. Moreover, in the later case, Assumption (H2) guarantees that a conflict can only occur near the common node. The following lemma takes care of the case where a pair of flights travel on arcs that are adjacent to a common node.

Lemma 1. *Let $u_1 = (v_1, w_1)$, $u_2 = (v_2, w_2)$ two arcs adjacent to a common node $v \in V$. Let θ_{u_1,u_2} be the angle between the vectors $\overrightarrow{v_1w_1}$ and $\overrightarrow{v_2w_2}$. Assume that flight f passes node v before g. Then, there is no conflict between f and g, when f is flying on u_1 while g is flying on u_2 if and only if*

$$t_g^v - t_f^v \geq \Delta_{u_1,u_2}^{f,g}, \tag{7}$$

where $\Delta_{u_1,u_2}^{f,g}$ is defined as follows:

1. *If $u_1 = (v_1, w_1)$ and $u_2 = (v_2, w_2)$ are converging arcs (i.e. $v = w_1 = w_2$), then*

$$\Delta_{u_1,u_2}^{f,g} := \begin{cases} \frac{d_h}{s_{u_2}^g} & \text{if } s_{u_1}^f \cos(\theta_{u_1,u_2}) \leq s_{u_2}^g \\ \frac{d_h\sqrt{(s_{u_1}^f)^2+(s_{u_2}^g)^2-2s_{u_1}^f s_{u_2}^g \cos(\theta_{u_1,u_2})}}{|\sin(\theta_{u_1,u_2})| s_{u_1}^f s_{u_2}^g} & \text{otherwise.} \end{cases}$$

2. *If $u_1 = (v_1, w_1)$ and $u_2 = (v_2, w_2)$ are serial arcs (i.e. $v = v_1 = w_2$), then*

$$\Delta_{u_1,u_2}^{f,g} := \begin{cases} \max\left(\frac{d^h}{s_{u_1}^f}, \frac{d_h}{s_{u_2}^g}\right) & \text{if } s_{u_1}^f \cos(\theta_{u_1,u_2}) \geq s_{u_2}^g \text{ or } s_{u_2}^g \cos(\theta_{u_1,u_2}) \geq s_{u_1}^f \\ \frac{d^h\sqrt{(s_{u_1}^f)^2+(s_{u_2}^g)^2+2s_{u_1}^f s_{u_2}^g \cos(\theta_{u_1,u_2})}}{|\sin(\theta_{u_1,u_2})| s_{u_1}^f s_{u_2}^g} & \text{otherwise.} \end{cases}$$

Consequently, the horizontal separation constraint on node v reads

$$t_g^v - t_f^v \geq \Delta_{u_1,u_2}^{f,g} - (1 - \delta_{f,g})M. \tag{8}$$

This constraint must be satisfied for each pair f, g of aircraft, for each node $v \in r_f \cap r_g$, and for each arcs $u_1 \in r_f$, $u_2 \in r_g$ adjacent to v.

3. Computational experiments

We test our approach on real traffic data sample recorded on 5th May 2015 at Paris CDG Airport on runway 26L. We apply our algorithm first at once on the 24-hour data and then on different 2-hour time windows. The MILP model is solved with Gurobi 5.6.3. Computations are performed on an Intel(R) Core(TM) i5-3210M with 2.5GHz and 4Go RAM memory. The results (number of non-achievable CTAs) are given in Table 1 and show that the problem can be efficiently solved.

Table 1. Results obtained on real traffic of the Paris CDG Airport runway 26L (May 5, 2015)

Time windows	0:00 - 24:00	3:00 - 5:00	5:00 - 7:00	7:00 - 9:00	9:00 - 11:00	11:00 - 13:00	13:00 - 15:00	15:00 - 17:00	17:00- - 19:00	19:00 - 21:00
Number of flights	417	32	63	39	51	54	42	48	48	37
Optimal solution	29	3	9	3	8	3	0	0	1	1
Computation time (sec.)	28.6	0.05	31.04	0.03	5.66	0.01	0.03	0.05	0.03	0.03

4. Summary

The problem of minimizing the number of CTAs falling outside the [ETA_{min},ETA_{max}] windows subject to operational constraints is investigated. A MILP formulation and promising preliminary computational experiments on real traffic data were presented.

References

[1] D. De Smedt, J. Bronsvoort, and G. McDonald. Controlled Time of Arrival Feasibility Analysis. Tenth USA/Europe Air Traffic Management Research and Development Seminar, Chicago, Illinois USA , June 2013.

[2] J.K. Lenstra, A.H.G. Rinnooy Kan, and P. Brucker. Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, 1:343-362, 1977.